Digital Optical Module
PLD Design

Lawrence Berkeley National Laboratory
4/5/99

Introduction

The PLD in the Digital Optical Module electronics acts as the glue logic between the CPU, its memory, the FPGA and the serial DACs and ADCs. Beside it holds the logic for a simple asynchronous serial protocol which is used by the DAQ to communicate with the CPU during the boot phase and low level software updates during which the packet protocol implemented in the FPGA communication controller is not available.

PLD properties

A Philips PZ312 type device is used. This PLD has 128 PLD cells, its operating voltage is 3.3V and it has a very low power consumption.

PLD configuration

The PLD is an EEPROM-type and can be configuered in system by JTAG either from a standard 10 pin JTAG header located on the DOM board or over the 50 pin test connector using the DOMTester. Any PC with a parallel port and the Philips configuration software can be used to configure the PLD during the DOM production. The PLD can **not** be reconfiguered by the DOM CPU because such a process would inevitably disrupt CPU operation since the PLD is used as the glue logic chip.

Design overview

The design can be partitioned into the following parts:

1. Glue logic
2. Register File
3. Ternary Signal Serial Communications CODEC

CPU/Memory/FPGA Glue Logic

The PS1711 CPU has six user chip select outputs CS0..CS5. They are mapped to six different memory areas in one of two different ways, depending on the boot mode of the CPU. In the external boot mode the CPU (the one used in the DOM) always boots a program at address 0 of memory block zero, i.e. a memory which is physically selected by CS0. This CS line therefore has to activate the system boot ROM after a **cold start** of the DOM. The system ROM is implemented as an 8 bit wide and 1 MByte deep OTP (one time programmable) EPROM. CS1 and CS2 can be used to access the system RAM and Flash memory respectively (each of which are 32 bit wide memories). CS3 is reserved for the FPGA address space (an eight bit memory mapped peripheral) while CS4 is used to access the PLD registers (which again is seen as an eight bit wide memory mapped peripheral device).

In order to boot an operating system residing in RAM or the Flash memory, the address scheme has to be altered by the boot code in such a way that CS0 is used as the RAM or

Flash chip select signal while the boot ROM is relocated to the address space covered by CS1 or CS2. These are called the **warm start modes**. The glue logic therefore implements the following mapping between the CPU's CSx outputs and the actual chip select signals ROMCS, RAMCS, FLASHCS, FPGACS and PLDCS:

	Cold start	Warm start RAM 0	Warm start RAM 1	Warm start FLASH 0	Warm start FLASH 1
CS0	ROMCS	RAMCS	RAMCS	FLASHCS	FLASHCS
CS1	RAMCS	ROMCS	FLASHCS	RAMCS	ROMCS
CS2	FLASHCS	FLASHCS	ROMCS	ROMCS	RAMCS
CS3	FPGACS	FPGACS	FPGACS	FPGACS	FPGACS
CS4	PLDCS	PLDCS	PLDCS	PLDCS	PLDCS

Table 1: CPU and memory/peripheral chip select signal map

The warm start mode can be controlled by the **Memory Configuration Register**. The actual memory configuration **does not change until a CPU reset is initiated**.

PLD Register File

The PLD has eight internal registers which are decoded by three address lines. Each register is up to eight bit wide. Some registers are read or write only, others can be written and read.

Address	Register name	Description
0w	FPGA Configuration Register Write - FPGAConfRegW	
0r	FPGA Configuration Register Read - FPGAConfRegR	
1w	ADC Control Register Write – ADCContrRegW	
1r	ADC Control Register Read – ADCContrRegR	
2w	DAC Control Register Write – DACContrRegW	
2r	DAC Control Register Read – DACContrRegR	
3w		
3r		
4w		
4r		
5w		
5r		
6w	Serial Communications Register	
6r	-	N/A
7w	Memory Configuration Register	
7r	-	N/A

Table 2: PLD Registers

The registers are controlling the following signals:

FPGA Configuration Register Write							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N/A	FPGA_TMS	FPGA_TDI	FPGA_TCK	N/A	nCONFIG	DATA0	DCLK

FPGA Configuration Register Read							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FPGA_TDO	N/A	N/A	N/A	CONF_DONE	N/A	N/A	N/A

Table 3: FPGA Configuration Register Bits

ADC Control Register Write							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DAC_CS3	DAC_CS2	DAC_CS1	DAC_CS0	DAC_CL	DAC_PDL	DAC_DIN	DAC_SCLK

DAC Control Register Read							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DAC_DOUT3	DAC_DOUT2	DAC_DOUT1	DAC_DOUT0	N/A	N/A	N/A	N/A

Table 4: DAC Control Register Bits

ADC Control Register Write							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N/A	N/A	N/A	N/A	ADC_STRB	ADC_CS	ADC_DIN	ADC_SCLK

ADC Control Register Read							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N/A	N/A	N/A	ADC_DOUT	N/A	N/A	N/A	N/A

Table 5: ADC Control Register Bits

Ternary Signal Serial Communications CODEC

In order to communicate with the CPU without having to load a communications design into the FPGA, a simplified (but slow) serial protocol is used. By encoding the binary serial signal of the CPUs UARTs with a ternary pulse code, the communication signal can be made DC-free, thereby satisfying the limitations of the ac coupled analog signal path.

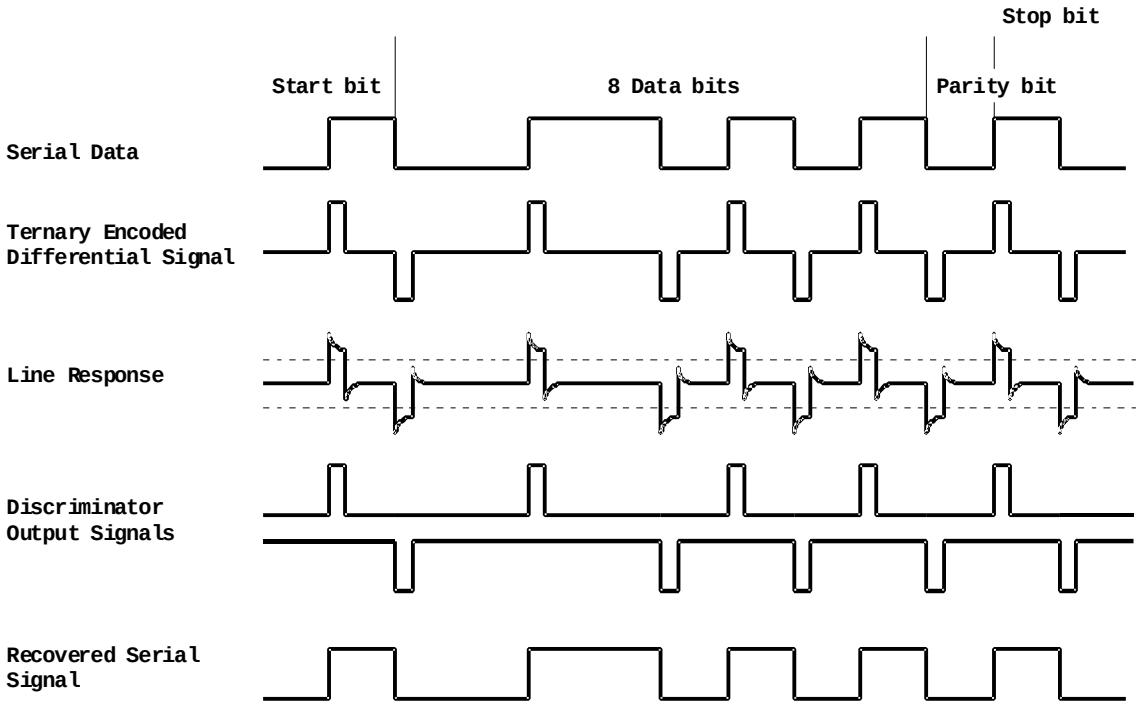

Diagram 1: Ternary Encoded Differential Serial Protocol

The implementation of the CODEC is relatively simple since the coder does not require more than two digital edge detectors and a timer circuit (made out of a simple counter), while the decoder is a simple RS-type flipflop which is triggered by two discriminator signals.